THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Exact inference in Bayesian networks and applications in forensic statistics

IVAR SIMONSSON

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
AND UNIVERSITY OF GOTHENBURG
Göteborg, Sweden 2018

Exact inference in Bayesian networks and applications in forensic statistics
IVAR SIMONSSON

ISBN 978-91-7597-818-5

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4499
ISSN 0346-718X

Department of Mathematical Sciences
Chalmers University of Technology
and University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 10 00
Author e-mail: simonssi@chalmers.se

Typeset with LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2018

Exact inference in Bayesian networks and applications in forensic statistics

Ivar Simonsson

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

Bayesian networks (BNs) are commonly used when describing and analyzing relationships between interacting variables. Approximate methods for performing calculations on BNs are widely used and well developed. Methods for performing exact calculations on BNs also exist but are not always considered, partly because these methods demand strong restrictions on the structure of the BN. Part of this thesis focuses on developing methods for exact calculations in order make them applicable to larger classes of BNs. More specifically, we study the variable elimination (VE) algorithm, which traditionally can only be applied to finite BNs, Gaussian BNs, and combinations of these two types. We argue that, when implementing the VE algorithm, it is important to properly define a set of factors that represents the conditional probability distributions of the BN in a suitable way. Furthermore, one should strive for defining this factor set in such a way that it is closed under the local operations performed by the algorithm: reduction, multiplication, and marginalization. For situations when this is not possible, we suggest a new version of the VE algorithm, which is recursive and makes use of numerical integration. We exemplify the use of this new version by implementing it on Γ-Gaussian BNs, i.e., Gaussian BNs in which the precision of Gaussian variables can be modeled with gamma distributed variables.

Bayesian networks are widely used within forensic statistics, especially within familial relationship inference. In this field, one uses DNA data and knowledge about genetic inheritance to make calculations on probabilities of familial relationships. When doing this, one needs not only DNA from the people to be investigated, but also data base information about population allele frequencies. The possibility of mutations makes these calculations harder, and it is important to employ a reasonable mutation model to make the calculations precise. We argue that many existing mutation models alter the population frequencies, which is both a mathematical nuisance and a potential problem when results are interpreted. As a solution to this, we suggest several methods for *stabilizing* mutation models, i.e., tuning them so that they no longer alter the population frequencies.

Keywords: Bayesian networks, exact inference, variable elimination, forensic statistics, familial relationship inference, mutation models

List of appended papers

Paper I **Simonsson, I.**, Mostad, P. (2016). Stationary mutation models. *Forensic Science International: Genetics, 23*, 217-225.

Paper II **Simonsson, I.**, Mostad, P. (2016). Exact Inference on Conditional Linear Γ-Gaussian Bayesian Networks. In *Conference on Probabilistic Graphical Models* (pp. 474-486).

Paper III **Simonsson, I.**, Mostad, P. A new algorithm for inference in some mixed Bayesian networks with exponential family distributions. *(Submitted)*

My contributions to the appended papers:

Paper I I co-developed and analyzed the matrix closeness criterion. I co-developed and implemented the stabilizations methods and applied them to data. I proved Theorem 1 and I did most of the writing for publication.

Paper II I worked out most of the formulas and helped adapt the VE algorithm to the framework of the Paper. I implemented this version of the algorithm and carried out the data analysis. I did most of the writing for publication.

Paper III I helped develop the ideas on the general level regarding the recursive VE algorithm and the prefamily VE algorithm. I co-developed the adaption to Γ-Gaussian BNs and I carried out the proofs. I implemented the algorithm and applied it to the examples. I did most of the writing for publication.

Acknowledgements

First and foremost I would like to thank my supervisor Petter Mostad for introducing me to new research topics and ideas and for teaching me so much during these years. You are one of the most kind-hearted people I have ever met and I don't want to imagine doing this without your support. I want to thank my co-supervisor Aila Särkkä for helping with my writing and for always being so nice and considerate. The department would be a much better place if more people were like you.

During my time at this department I have been surrounded by great people all the time. Special thanks to you Claes for being such a good friend during these years, I feel really fortunate to have had you around. Thank you Magnus, Peter, Dawan, Mariana, Henrike, Viktor, Fredrik, Malin, Anna, Mikael, Fanny, Olle, Jonathan, Marco, Tuomas, Maud, Tobias and many more, for making the department a great place to be.

Warm thanks goes out to all my friends from real life, for keeping in touch with me even though I show very little signs of trying to do the same. To the Oldtown *** crew, thanks for all the great hang outs, both online and offline. It feels like we have an unbreakable bond and I can't describe how happy that makes me. To my oldest friends, Joel and Björn, you are always amazingly fun to hang out with and I never feel so at home as when I hang out with you. Thank you Mattias, Jakob, Jonatan, Oscar and Bamme. Gracias a mis amigos madrileños! El año en Madrid fue uno de los mejores años de mi vida. Siendo guiri, no puedo esperar para hacerme sueco con vosotros otra vez.

Thank you mom and dad for always being there for me. Your constant support means the world to me and I don't know who I would be without it. Thank you Ida and Hanna, you are hands down the two people I look up to the most in life and I wish I could be more like you.

Finally, thank you Sandra for being you and letting me be me, and for all support and comfort you have given me this past year.

Contents

Chapter 1

Introduction

Bayesian networks can be used in a wide variety of applications and the intuitive way they can be constructed makes them suitable for modeling large classes of problems. Inference on Bayesian networks is less intuitive and is often performed approximately using simulation methods. There is also software that can perform exact inference on Bayesian networks, both for general use, for example HUGIN[1] and GeNIe[2], and for more subject specific use, for example Familias[3]. However, algorithms for exact inference are limited to rather narrow subclasses of Bayesian networks. One of the main themes in this PhD project has been to extend the classes of Bayesian networks for which the existing algorithms can be implemented. Another important theme has been familial relationship inference, a subfield of forensic statistics. Therefore, it feels natural to include the following three parts in this thesis: Bayesian networks and exact inference, forensic statistics, and familial relationship inference.

In Chapter 2 we first introduce Bayesian networks and, following [11], we then continue by describing the *variable elimination algorithm*, which is a general tool for performing exact inference on Bayesian networks. The idea behind this algorithm is to identify smaller components of the network, associated with the so-called *factors*, and define local operations on these, instead of considering the whole network at once. Although the algorithm is presented in a very general setting, one has to study the structure of the factors more carefully to be able to use it in practice. The way the factors of the network are represented must realistically be unified in order to make implementations possible. Meanwhile, the form of the factors depends on the distributions of the random variables in the network.

In Chaper 3 we give a brief introduction to forensic statistics, mainly based

[1] http://www.hugin.com/
[2] http://www.bayesfusion.com/
[3] http://familias.no/english/

on [14]. As for Bayesian networks in general, it is often intuitive to formulate the structure of the networks representing legal cases. However, which variables to include in the network, and how to define them properly, is worth investigating carefully. Generally, it is desirable to include many variables in the network in order to make the interaction between them less complicated, rather than to have a small network with interactions that are hard to specify. Moreover, one often needs to collaborate with field experts in order to define the conditional probability distributions accurately before applying the theory in Chapter 2 to perform calculations.

Chapter 4 is a brief introduction to the area of familial relationship inference from DNA data. This can be seen as a subfield of general forensic science, and the network formulations can be recognized from the previous chapters. When making inference within this area, there are a lot of complicating issues that need to be accounted for in the model formulation as a whole, see [10]. One of these issues, namely the possibility of mutations, is the focus of Paper I, hence it is also described in detail in Chapter 4.

Chapter 2

Bayesian networks

A pioneering work and standard reference within the theory of Bayesian networks is the book *Probabilistic Reasoning in Intelligent Systems* from 1988 by Judea Pearl, [13]. Later key references include [5] and [11], the latter being an inspiring source for the notations and structure of this chapter. In what follows we will give an introduction to Bayesian networks and how one can perform exact inference on them.

2.1 Terminology and notation

A *graph* $\mathcal{G} = (V, E)$ consists of a collection of nodes (or vertices) V and a collection of edges E. Each edge in E connects two nodes in V and if these nodes are the same, the edge is said to be a *loop*. An edge pointing from one node to another is said to be *directed*, and if E only consists of directed edges, then $\mathcal{G}$ is said to be a directed graph. A *directed path* in $\mathcal{G}$ is a series of alternating nodes and edges $v_1e_1v_2e_2\cdots v_{n-1}e_{n-1}v_n$, such that e_i points from v_i towards v_{i+1}, for all i. If $v_1 = v_n$, then this path is called a *cycle*. *Directed acyclic graphs* (DAGs), i.e., directed graphs with no cycles, are of particular interest to us. Note that a loop is actually a cycle, so a DAG can not have loops. From now on, all graphs we consider will be DAGs.

A node $v \in V$ from which there is an edge pointing towards another node $u \in V$ is called a *parent* of u, and the set of parents of u is denoted by $\mathrm{Pa}_u^{\mathcal{G}}$. Often it is clear what the underlying graph is, and we omit the superscript $\mathcal{G}$.

We will use calligraphic letters to denote sets of random variables, e.g., $\mathcal{X} = \{X_1, \ldots, X_n\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$, and we will use boldface for random vectors, e.g., $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_m)$. We can perform set operations, for example $\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$ and $\mathcal{Z} = \mathcal{X} \setminus \mathcal{Y}$, which by extension defines a random vector $\mathbf{Z}$ form $\mathbf{X}$ and $\mathbf{Y}$. We will by $Val(X)$ denote the range of a random variable X, i.e., the set of possible values the variable can attain.

Similarly, for a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ we use $Val(\mathbf{X})$ to denote the product set $Val(X_1) \times \cdots \times Val(X_n)$.

We will study graphs whose nodes represent random variables. We say that $\mathcal{G} = (V, E)$ is a graph *over* a set of random variables $\mathcal{X}$ if each node in V represents a single random variable in $\mathcal{X}$, and if each random variable in $\mathcal{X}$ is represented by a single node in V. Even though it is somewhat dubious, we will write X when referring both to the random variable X and to the node in the graph that represents X. Moreover, when we write Pa_X we will mean the random variables that are represented by the parents of the node in the graph that represents X.

A *Baysian network* (BN) over a set of random variables $\mathcal{X}$ consists of a DAG $\mathcal{G}$ whose nodes correspond to variables in $\mathcal{X}$, together with a probability distribution over $\mathcal{X}$, whose density fulfills

$$\pi(X_1, \ldots, X_n) = \prod_{i=1}^{n} \pi(X_i | \mathrm{Pa}_{X_i}). \tag{2.1}$$

Equation (2.1) is called the *chain rule* for Bayesian networks and the individual factors on the right-hand side are the *conditional probability distributions* (CPDs) of the network. Note that the information contained in the CPDs is the only information needed to recreate the BN since the network structure is implicit: we draw an arrow from X_i towards X_j if and only if $X_i \in \mathrm{Pa}_{X_j}$.

2.2 The variable elimination algorithm

Using Equation (2.1), we can reasonably effectively perform exact computations on Bayesian networks. For example, if we have a BN over a set of random variables $\mathcal{X}$ and we have evidence $\mathbf{y}$ on some vector $\mathbf{Y}$ whose variables are in $\mathcal{X}$, then the conditional density $\pi(X|\mathbf{Y} = \mathbf{y})$ for some variable $X \in \mathcal{X} \setminus \mathcal{Y}$ can be computed with the *variable elimination* (VE) algorithm. The relevant objects that are being used in this algorithm are called *factors*. A factor over a set of random variables is simply defined as a non-negative real valued function on the range of the variables.

Definition 1. A *factor*, ϕ, over a set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ is a function from $Val(\mathbf{X})$ to $\mathbb{R}_{\geq 0}$. The set $\mathcal{X}$ is called the *scope* of ϕ.

Initially, before running the algorithm, the factors will consist of the CPDs of the given network. The algorithm will then sequentially perform a series of operations on these factors. If we have evidence about some variables in the network, the first step is to introduce this evidence. Within the algorithm, this is called *factor reduction* and is formally defined as follows.

Definition 2 (Factor reduction)**.** Let ϕ be a factor over a set of variables $\mathcal{X}$ and assume that we have evidence on some other set of variables $\mathcal{Y} = \{Y_1, \ldots, Y_m\} \subseteq \mathcal{X}$, so that $\mathbf{Y} = \mathbf{y}$ for some $\mathbf{y} \in Val(\mathbf{Y})$. The *factor reduction* of ϕ with respect to the evidence $\mathbf{Y} = \mathbf{y}$ is a new factor ϕ' over $\mathcal{Z} = \mathcal{X} \backslash \mathcal{Y}$ such that $\phi'(\mathbf{z}) = \phi(\mathbf{z}, \mathbf{y})$ for all $\mathbf{z} \in Val(\mathbf{Z})$. The new factor ϕ' is sometimes denoted by $\phi[\mathbf{Y} = \mathbf{y}]$.

Note that a factor is just a function of multiple variables. One way to interpret factor reduction is that we are fixing a subset of the variables of a multivariate function.

From Definition 2, we see that if $\mathcal{Y} = \mathcal{X}$, the resulting factor is simply a real number and its scope will be the empty set. It will later be clear that a factor with empty scope only affects the rest of the algorithm by a scaling factor and since we are dealing with probabilities we might as well perform this scaling in the end of the algorithm with a normalization step. Hence, if we are about to reduce a factor with respect to its entire scope, we simply remove the factor instead.

After performing factor reduction on each affected factor, it is time to start eliminating variables. Eliminating a variable X consists of two major steps. The first step consists of multiplying all factors whose scope includes X, and in the second step this product is marginalized in order to obtain a factor that does not include X, and hence X is eliminated. Mathematically, we can describe these steps by

$$\phi' = \int_{Val(X)} \left(\prod_{\phi \in \Phi'} \phi \right) dX,$$

where Φ' is the set of factors whose scope contains $\mathcal{X}$. Note that if X is discrete, the integral is a sum.

Formally, we define two operations to perform this variable elimination, namely *factor multiplication* and *factor marginalization.*

Definition 3 (Factor multiplication)**.** The *factor multiplication* of two factors, ϕ_1 over $\mathcal{X}$ and ϕ_2 over $\mathcal{Y}$, is a new factor ϕ over $\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$ such that $\phi(\mathbf{z}) = \phi_1(\mathbf{x})\phi_2(\mathbf{y})$ for all $\mathbf{z} \in Val(\mathbf{Z})$.

Definition 4 (Factor marginalization)**.** Let ϕ be a factor over $\mathcal{Z}$ and let $\{\mathcal{X}, \mathcal{Y}\}$ be a partition of $\mathcal{Z}$. The *factor marginalization* of ϕ with respect to $\mathcal{Y}$ is another factor ϕ' over $\mathcal{X}$, such that

$$\phi'(\mathbf{X}) = \int_{Val(\mathbf{Y})} \phi(\mathbf{X}, \mathbf{Y}) d\mathbf{Y} \tag{2.2}$$

whenever this integral exists. When performing factor marginalization, we will say that we *marginalize out* $\mathbf{Y}$ from ϕ.

Again, if $\mathbf{Y}$ is discrete in (2.2), the integral is a sum. In fact, $\mathbf{Y}$ could also be partly discrete and partly continuous, in which case (2.2) should be interpreted accordingly. The discussion succeeding Definition 2 also applies here, hence if $\mathcal{X} = \emptyset$ in Definition 4, we simply remove the resulting factor.

The operations of Definitions 2-4 will be called the *local operations*, since they act locally on the network. With help of these operations we can formalize the VE algorithm. As input, we need to specify the BN and all its CPDs, and we also need to specify the query variables, i.e., the variables (usually only one) whose distributions we want to compute. As optional input we can specify evidence, i.e., observed variables. Variables that are neither observed nor part of the query are called *elimination variables* because we will eliminate them all, one by one. In Algorithm 1 below we present the VE algorithm in pseudo code.

Algorithm 1 VariableElimination(Π,$\mathcal{X}$,$\mathbf{y}$)

Require: A set Π of CPDs of the BN, query variables $\mathcal{X}$, and evidence $\mathbf{y}$
1: Construct factors $\Phi = \{\phi_1, \ldots, \phi_N\}$ from the CPDs in Π
2: Partition the set of variables into query $\mathcal{X}$, evidence $\mathcal{Y}$ and elimination $\mathcal{Z}$
3: **for all** $i = 1, \ldots, N$ **do**
4: Replace ϕ_i by $\phi_i[Y = y]$
5: **end for**
6: Fix an ordering $Z_1, \ldots, Z_k$ of the variables in $\mathcal{Z}$
7: **for** $i = 1, \ldots, k$ **do**
8: $\Phi' \leftarrow \{\phi \in \Phi : Z_i \in \text{Scope}(\phi)\}$
9: $\Phi'' \leftarrow \Phi \setminus \Phi'$
10: $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$ {factor multiplication}
11: $\rho \leftarrow \int_{Z_i} \psi(\cdot) dZ_i$ {factor marginalization}
12: $\Phi \leftarrow \Phi'' \cup \{\rho\}$
13: **end for**
14: **return** Φ

2.3 Existing work – a few special cases

A big challenge in implementing the VE algorithm lies in how to best represent the factors we need. We will are not able to implement the algorithm for the most general BNs in which the only restriction on the CPDs is that they have to be probability distributions. We simply can not represent this level of generality in implementations, and therefore we put restrictions on the BNs.

Let us start by looking at finite networks, i.e., networks in which all variables have a finite range, where we can represent all CPDs, and thus all initial factors in the algorithm, as finite vectors. It is not too hard to conclude that all factors that show up later in the algorithm are also representable by finite vectors. This

follows from the fact that the set of finite vectors is closed under the local operations, i.e., applying any of the operations in Definitions 2-4 on finite vectors results in a finite vector. So, it is possible to implement Algorithm 1 so that it works for *all* finite BNs (at least assuming unlimited space and time). There exists some software for this, for example both the aforementioned HUGIN and GeNIe can handle this.

2.3.1 Gaussian Bayesian networks – canonical forms

Apart from finite BNs, the "golden" class of networks for which the VE algorithm is implementable is the class of *Gaussian* Bayesian networks.

Definition 5. We say that a Bayesian network over $\mathcal{X} = \{X_1, \ldots, X_n\}$ is *Gaussian* if for each $i = 1, \ldots, n$, we have that $X_i|\mathrm{Pa}_{X_i} \sim \mathcal{N}(\mu_i, \sigma_i^2)$, where

$$\mu_i = \alpha + \sum_{k=1}^{m} \beta_k Y_k.$$

Here $Y_1, \ldots, Y_m$ are the parents of X_i and $\sigma_i^2, \alpha, \beta_1, \ldots, \beta_m$ are all real valued constants.

A good way to implement the algorithm for Gaussian BNs is to use a factor type that is called *canonical forms* in [11] and conditional Gaussian (CG) potentials in [5]. We will use the former name.

Definition 6. Let $\mathbf{X}$ be a random vector and let ϕ be a factor over $\mathcal{X}$. We say that ϕ is a *canonical form*, denoted by $\mathcal{C}(\mathbf{X}; K, \mathbf{h}, g)$ (or simply $\mathcal{C}(K, \mathbf{h}, g)$), if it can be written as

$$\phi(\mathbf{X}) = \mathcal{C}(\mathbf{X}; K, \mathbf{h}, g) = \exp\left(-\frac{1}{2}\mathbf{X}^T K \mathbf{X} + \mathbf{h}^T \mathbf{X} + g\right) \tag{2.3}$$

where $K \in \mathbb{R}^{n \times n}$ is symmetric, $\mathbf{h} \in \mathbb{R}^n$ and $g \in \mathbb{R}$.

As presented in [11], the density of a normally distributed vector, $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$, can be written on canonical form with

$$\begin{cases} K &= \Sigma^{-1} \\ \mathbf{h} &= \Sigma^{-1}\mu \\ g &= -\frac{1}{2}\mu^T \Sigma^{-1}\mu - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma|. \end{cases} \tag{2.4}$$

This relation is not hard to prove and can be hinted at by observing that both the Gaussian density and the canonical forms are natural exponentials of a quadratic polynomial. In fact, there is a duality between the Gaussian

density and a subset of canonical forms. As seen in Definition 6, there is a constant parameter g in the parameterization of canonical forms and there is no restriction that a canonical form has to integrate to one, or even to a finite number. We have the following proposition, which will be useful later on.

Proposition 1. *If $\mathbf{X} = (X_1, \ldots, X_n)$ is a normally distributed random vector with mean μ and covariance matrix Σ, then its density is a canonical form over $\mathcal{X}$, with parameters given by (2.4). Conversely, if $\phi(\mathbf{X}) = \mathcal{C}(\mathbf{X}; K, \mathbf{h}, g)$ is a canonical form over a random vector $\mathbf{X}$, with K positive definite, then $\phi(\mathbf{X})$ is proportional to the normal density for $\mathbf{X}$, with $\Sigma = K^{-1}$ and $\mu = K^{-1}\mathbf{h}$.*

The VE algorithm can be smoothly implemented for finite BNs, since the set of finite dimensional vectors is closed under the local operations. Except for some cases of marginalization, which we will be able to avoid, the set of canonical forms is also closed under the local operations, hence we are able to implement the VE algorithm for these networks as well.

2.3.2 Mixed networks

So far we have seen that the VE algorithm is implementable for finite BNs and for Gaussian BNs. A natural next step is to investigate whether it also works on networks in which some nodes have a finite range and the rest are normally distributed. After all, the construction of such Bayesian networks should be straightforward. It turns out that this is indeed possible if we impose the restriction that finite variables have exclusively finite parents and that the CPDs of each Gaussian variable is, for a fixed configuration of its finite parents, as in Definition 5. We use the name *mixed networks* for the networks that obey these restrictions.

The type of factors that show up when implementing the VE algorithm for mixed networks are the following.

Definition 7. Let ϕ be a factor over a set of random variables $\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$, where each $X \in \mathcal{X}$ is finite, and each $Y \in \mathcal{Y}$ is continuous. We say that ϕ is a *canonical table* if for each $\mathbf{x} \in Val(\mathbf{X})$, it can be written as

$$\phi(\mathbf{x}, \mathbf{Y}) = \sum_{i=1}^{n} \phi_i(\mathbf{Y}), \tag{2.5}$$

where each $\phi_i(\mathbf{Y})$ can be written in the form (2.3).

The choice of using the word *table* is motivated by the intuitive way of interpreting these factors. For each configuration of the finite variables, we have an entry (which is just an element of the corresponding finite vector if the factor is completely finite) and each entry is a sum of canonical forms. If one does

not find this intuition appealing, a canonical table could equally well be seen as simply a collection of functions of the form (2.5), where each such function is connected to a configuration of the finite variables in the scope of the factor. Nonetheless, we will continue to use the word table.

The local operations on canonical tables will be different depending on the circumstances. For example, factor marginalization will be performed differently depending on what kind of variable, finite or continuous, we are about to marginalize out. The resulting factor will also look slightly different. When constructing the initial canonical tables from a mixed Gaussian BN, we will always have $n = 1$ in (2.5). However, when we marginalize finite variables out of the factor, n will increase. We present a more general approach for mixing continuous and finite variables in Paper III.

In [5], it is suggested that instead of allowing sums of canonical forms in the canonical tables, the marginalization operation should be done approximately, using so called *weak* marginalization. Using Proposition 1, one identifies a Gaussian density with each term in the resulting sum of factors. It is then possible to produce the mean and the covariance matrix of the random vector that this sum is a distribution for. Even though this random vector is not Gaussian, one returns the density of a Gaussian random vector with the produced mean and covariance matrix, which is reformulated to a canonical form again using Proposition 1.

We will refrain from explicitly specifying the local operations on canonical tables, partly because they are already determined implicitly by Definitions 2- 4, and partly because it is notationally rather messy.

2.4 An extension

As mentioned earlier, if we want to implement the VE algorithm for a specific class of BNs, it is very convenient to find a corresponding factor class that is closed under the local operations. Unfortunately, it turns out that the examples given in Section 2.3 are, at least to our knowledge, the only cases where this has been done in a general way.

An interesting question is therefore, whether we can do something else when closedness can not be achieved. In general, for a particular class of BNs, we could attempt to proceed as follows.

1. Create a set of factors that includes all possible CPDs that can occur in this class.

2. Check whether this set also includes all factors that will be created within the algorithm, and if

   a) yes, go to Step 3.

b) no, extend the factor set to also include the factors that could be created within the algorithm, and go back to Step 2.

3. Implement the algorithm for the current set of factors.

In the completely finite and Gaussian cases, we would reach Step 2a directly and never reach Step 2b.

In Paper II, we introduce a class of BNs that is based on a Gaussian BN but has one additional variable: a gamma distributed variable to model the precision of the Gaussian variables in the network. Applying Step 1 to this BN, we see that both the gamma variable (denoted by τ below) and the Gaussian variables have CPDs that can be written in the form

$$\exp\left(\tau\left(-\frac{1}{2}\mathbf{X}^T K\mathbf{X} + \mathbf{h}^T X + g\right) + a\log(\tau) + b\right). \tag{2.6}$$

It turns out (see the details in Paper II) that the factor set defined as the set of factors that can be written in the form (2.6) is closed under factor reduction and factor multiplication but not quite closed under factor marginalization. While marginalizing such a factor with respect to a variable in X leads to a factor within this set, marginalization with respect to the gamma distributed variable τ leads to a factor outside this set. Therefore, we end up in Step 2b above. Trying to extend the factor set and repeat this process will lead to more complications. The extended factor set will be of much more complicated structure than (2.6) and it will not be closed under the local operations.

Unfortunately, this is generally what happens when one tries to apply the rather naïve thought process presented in Steps 1-3 above. The problem is that when one tries to extend the factor set, the structure of the factors becomes too complicated and the parameters too many and it ends up being unfeasible to represent the factors in code. It seems that we not only want the factor set to be closed under the local operations but we also want the factor structure and parameterization to be compact enough so that we are able to store and manipulate the factors in a computer.

In Paper II, we solve this problem by defining our factor set as the factors that can be written in the form (2.6). We then make sure that the marginalization that takes us out of this set, i.e., the marginalization w.r.t. τ, is done in the end, after all other marginalizations. This is possible since there is only one gamma variable, so there is only one problematic marginalization.

The ad hoc type of solutions in Paper II are, in our experience, what needs to be done when closedness under the local operations is not achieved for particular classes of BNs. In Paper III, we have a more general approach and we adopt a more thorough procedure than the thought process with Steps 1-3 above, namely we introduce the notion of *families* and *prefamilies*. Families are factor sets that are closed under all three local operations while prefamilies are closed

under factor reduction and factor multiplication but not necessarily under factor marginalization. We then define a new, recursive variable elimination algorithm that uses numerical integration instead of marginalization in the case when marginalization would take us out of the prefamily. At first it may seem that this approach would lead to computations of high-dimensional integrals, but, as exemplified in Paper III, this can sometimes be avoided thanks to the recursive nature of the algorithm.

Chapter 3

Forensic statistics

In forensic science we are often concerned with assessing how some particular evidence, E, influences a legal case. If the case is two-sided, i.e., if there are two competing hypotheses, H_1 and H_2, we ultimately want to consider the relationship between the probabilities of each hypothesis, after taking the evidence into account. More precisely, we consider the posterior odds, i.e., the ratio between $\Pr(H_1|E)$ and $\Pr(H_2|E)$. A key mathematical tool in order to produce the posterior odds is Bayes' rule on odds form, which says that the posterior odds is equal to the likelihood ratio times the prior odds, or more formally:

$$\frac{\Pr(H_1|E)}{\Pr(H_2|E)} = \frac{\Pr(E|H_1)}{\Pr(E|H_2)} \times \frac{\Pr(H_1)}{\Pr(H_2)}. \tag{3.1}$$

As we can see, we need both the likelihood ratio and the prior odds in order to compute the posterior odds. The prior odds is in general affected by various circumstantial factors and it is usually up to the legal experts to produce it. The forensic scientist is left with what can be affected by the data, i.e., the likelihood ratio.

Example 1. Consider a case in which the prosecution claims that a particular man has committed a burglary, but the defense claims that he is innocent. Formally we have

H_p: The man committed the burglary

H_d: The man is innocent

where the subscripts represent the prosecution and the defense, respectively. A broken window is found at the crime scene and glass fragments are found on the jacket of the suspect.

This is typically the point where the forensic scientist is consulted: a suspect is identified and some evidence has been collected that could possibly tie the suspect to the crime scene. Using (3.1), we want to update our beliefs in the hypotheses via the likelihood ratio, which could be computed as

$$\frac{\Pr(E|H_p)}{\Pr(E|H_d)} = \frac{\Pr(E_c, E_s|H_p)}{\Pr(E_c|H_d)\Pr(E_s|H_d)}. \tag{3.2}$$

Here we have split up the evidence into E_c and E_s, which denote the evidence from the crime scene and from the suspect, respectively. We have also used that

$$\Pr(E_c, E_s|H_d) = \Pr(E_c|H_d)\Pr(E_s|H_d)$$

which is true since E_c and E_s are independent if the suspect is innocent[1]. Moreover, in this example the evidence from the crime scene is not dependent on which hypothesis is true, hence $\Pr(E_c|H_p) = \Pr(E_c|H_d)$. This will convert (3.2) into

$$\frac{\Pr(E|H_p)}{\Pr(E|H_d)} = \frac{\Pr(E_s|E_c, H_p)}{\Pr(E_s|H_d)} \tag{3.3}$$

which is ultimately the way we compute the likelihood ratio.

So we have to produce two probabilities, $\Pr(E_s|H_d)$ and $\Pr(E_s|E_c, H_p)$. As long as the hypotheses are formulated on the source level (see Secion 3.1 below), we usually estimate $\Pr(E_s|H_d)$ by doing comparisons towards data bases to see how common this type of evidence is in general. The value of $\Pr(E_s|E_c, H_p)$ on the other hand, is a measure of similarity between E_c and E_s, and the estimation procedure will be different depending on what kind of evidence we have. When glass evidence is collected, as in Example 1, one could measure the *refractive index* (RI), which is a measure of the optical density of the glass, see for example [15] (there are also other methods for analyzing glass data, see for example [6]). We then have to estimate the probability of observing E_s given the type of glass found on the crime scene and given that the suspect is guilty.

The evidence E_c and E_s above are named after their source, i.e., the crime scene and the suspect. Evidence is sometimes classified into *control evidence*, whose source is known, and *recovered evidence*, whose source is unknown. In the example above, E_c would be control evidence and E_s recovered evidence. However, it is not always the case that the control evidence comes from the crime scene and the recovered evidence from the suspect. If instead of a broken window, we had found blood stains on the crime scene, then these blood stains will constitute the recovered evidence and blood samples taken from the suspect would be the control evidence. There are also other classifications of evidence, see [1] for a more thorough discussion on this.

[1] Actually, in some cases E_c and E_s are only approximately independent given H_d, but for simplicity we do not elaborate more on that now.

Before proceeding, we realize that we also have to take into account case specific background information that is not directly related to the evidence. This can include, for example, eyewitness testimonies and relevant information about the suspect, and we will denote this general collection of background information by I. If we incorporate I into (3.3) we get

$$\frac{\Pr(E|H_p, I)}{\Pr(E|H_d, I)} = \frac{\Pr(E_s|E_c, H_p, I)}{\Pr(E_s|H_d, I)}.$$

3.1 Proposition levels

The hypotheses H_p and H_d above are formulated solely with the court's decision in mind. However, it is not always appropriate for the forensic scientist to consider propositions of this kind directly, simply because the evidence at hand is not directly connected to guilt. In [4], Cook et al. introduced three levels for propositions: *source level* (level 1), *activity level* (level 2) and *offense level* (level 3). The hypotheses introduced in Example 1 are on the offense level because they are statements about the guilt of the suspect. We could reformulate these hypotheses in the following way.

H_p: The suspect broke the window at the crime scene

H_d: The suspect did not break the window at the crime scene

These are statements about the breaking of the glass, hence they are on the activity level. We could also formulate hypotheses on the source level:

H_p: The glass fragments on the jacket come from the broken window at the crime scene

H_d: They come from some other source

It is tempting to say that these three pairs of hypotheses will give rise to the same conclusions and that it does not matter which pair we address. To argue that the choice of proposition level matters, consider first the difference between the offense level and the activity level propositions in this case. It is possible that the suspect committed the burglary even though someone else (or something else) broke the window. He could have had an accomplice who broke the window, or the window could have already been broken when the suspect arrived at the crime scene. On the other hand, it is also possible, although maybe more far-fetched, that the suspect did not commit the burglary even though he broke the window. The suspect might have been the accomplice who broke the window in order to then leave the crime scene. However, this last example would imply that both hypotheses at the offense level are false.

Maybe even more apparent in this case is the difference between the activity level and the source level. The suspect might be a suspect because he is known by the police from previous crimes and he might frequently come into contact with broken glass. Even if the glass from the suspect does not come from the broken window, he might have broken the window by for example throwing a rock through it. On the other hand, if the glass fragments on the jacket come from the window, they could have been planted there by the real offenders.

In general, the forensic scientist will have more authority regarding propositions on the source level, simply because the statements in these propositions are more closely related to their expert knowledge. Often it requires additional information and assumptions to be able to answer questions about guilt. This is a dilemma since the court is ultimately concerned with the offense level propositions. In order to answer questions on the offense level, we need to first answer questions on the source level, then consider how other circumstances affect the answer if the level is raised.

One example of circumstances that might affect this is who the suspect is and why he has become a suspect. Was he found in connection to the crime scene at the approximate time of the crime, or was he found in the same neighborhood two days later, or is he a suspect because of other related crimes he has committed? Another example might be what we know about the burglary. Are there witnesses who can confirm how the window was broken or do we even know that the window was broken by a person? It seems that a forensic scientist needs to be careful not to draw conclusions based on circumstances that are outside the area of expertise. A more general discussion of this major dilemma can be found in [4], which also includes a more detailed discussion about the particular glass example.

3.2 Bayesian networks in Forensic Science

The dilemma of using the correct level on the propositions could be broken down into smaller units. A natural approach is to start with modeling the source level propositions properly, and then investigate how various factors could affect the conclusions if we raise the proposition level. If we want to model this in a Bayesian network, we could introduce variables on several proposition levels. Consider the following example, introduced in [14].

Example 2. A single blood stain is found on a crime scene and there is a suspect from whom a blood sample is taken. As discussed above, the court is ultimately concerned with propositions on the offense level, hence the hypotheses could initially be that the suspect committed the crime (H_p) and that the suspect is innocent (H_d). We should have a variable for this in the network, call it H. The values of this variable should indicate which hypothesis is true, for example

$H = 0$ corresponds to H_p being true and $H = 1$ corresponds to H_d being true. Since we are ultimately interested in the hypotheses on this level, it is the node H that will be our query node. Which other variables do we want to include in the Bayesian network that models the case? Naturally it makes sense to have a node for the evidence, call it E. Could we add an edge directly from H to E? Probably, but it could be complex to specify the conditional distribution of E given H, there are simply too many factors that will affect this. On the other hand, the question that the forensic scientist really can answer is whether or not the blood stain was left by the suspect. So let's add a node for this uncertainty, call it F. Moreover, we could wonder if the blood stain was left by the offender. This is important because only if the blood stain was left by the offender does it make sense to consider the blood stain as evidence for guilt. Let G denote the node that represents whether or not the blood stain was left by the offender.

Now we consider edges again. It should be clear that F depends on H since it is less likely that the suspect left the blood stain if he is innocent. On the other hand, the guilt of the suspect should not be affected by the activity of the offender, unless we can make connections between the suspect and the offender. This implies that H and G are independent, but that they are dependent conditionally on F. This is achieved if G is a parent of F. Finally, it is clear that the evidence E should only have F as its parent. The resulting graph can be seen in Figure 3.1.

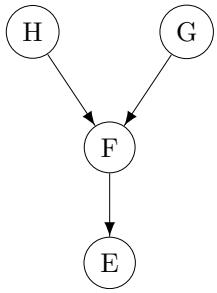

Figure 3.1: A graph visualizing the dependencies between the variables in Example 2.

The networks is very small and the calculations in Example 2 are done by hand in [14]. The prior probability that the blood stain at the crime scene came from the offender, i.e., the distribution of G, should arguably be left to the court experts to decide upon. Specifying the distribution of H could be done in two ways. The first option is to leave this problem to the court as well, use their prior for the distribution of H, and then compute the posteriors directly with our algorithm. The second option is to simply put a $Bernoulli(p = 0.5)$ prior on H, which means that our algorithm will produce the LR, and we have to account

for the prior for H separately. What is left implicit in the formulation of the network of Example 2 is how we compute the probability that the crime scene blood stain and the blood sample of the suspect comes from the same source. In ideal situations this is very easy since DNA profiles are almost unique. However, while the quality of the blood sample from the suspect will be almost flawless, this is seldom true for the blood stain from the crime scene. In reality this needs to be handled properly, see for example [3].

The introduction of the node G in Example 2 could be seen as a way to model how background information, included in the letter I earlier, affects the network. Other nodes could also be added to incorporate I more explicitly. For example, if there is a possibility that there were more than one offender, we have to rethink the inclusion of the node G. Again, see [14] for more details.

A graph similar to the one in Figure 3.1 can also be made for the case in Example 1. Then it might be appropriate to have hypothesis nodes on all three activity levels, call them H_O, H_A, and H_S for offense, activity and source, respectively. If this is done, H_O should be a parent of H_A, which should be a parent of H_S. These hypothesis nodes could also be affected by the background information in I. For example, we could add a node as a parent of H_S representing the possibility that glass fragments were planted on the suspect's jacket by the real offenders. Or we could add a parent to H_A representing in what way the window was broken. In any case, we would want H_S to be a parent of the evidence node. Using the same idea as in the earlier discussion about Example 1, we split the evidence node into two, E_C and E_S, hence we want H_S to be a parent of both these nodes. As for the interaction between F and E in Example 2, the interaction between H_S and its children in this case has not been discussed much. Note that this interaction is probably the one in which most trust is put on the forensic scientist. While the specifications of the other parts of the network could be done in collaboration with the court, how the data affect the source level propositions should be left entirely to the forensic scientist.

So how is this important interaction modeled? So far we have only mentioned that we should consult some data bases, but we have not talked about how this can be done. In fact, this interaction could also be modeled with Bayesian networks.

In Paper II, we do this for glass evidence, i.e., refractive index measurements. Generally, when one measures the refractive index of glass, repeated measurements are done, hence we would get several observations from each source. We denote by $x_{C,j}$ and $x_{S,j}$ the j:th measurements of the glass source from the crime scene and the suspect, respectively. In this case, we also have n data base glass sources $i = 1, \ldots, n$, on each of which we have measurements x_{ij}. We then assume that the measurements from each glass source $i = 1, \ldots, n, C, S$ are normally distributed around some source specific means, $\theta_1, \ldots, \theta_n, \theta_C, \theta_S$. In

turn, these means are drawn from some normal prior θ. What we hope for with this model is to judge if the difference between the measurements $x_{C,j}$ and $x_{S,j}$ is larger than one would expect the difference within a group of measurements $x_{i1}, \ldots, x_{ik}$ to be. In Paper II, we introduce a gamma distributed variable to model the variance of all normally distributed variables in the network. The resulting network can be seen in Figure 2 in Paper II in which we also have a hypothesis node, H, as query node.

Chapter 4

Familial relationship inference

In Chapter 3 we discussed the role of a forensic scientist in general, and the use of Bayesian networks as a tool to break down and to understand complicated circumstances. In this chapter, we will focus on an important type of forensic cases, namely familial relationships. In Paper I, possible solutions to a particular problem that arises from the mathematical tools used in familial relationship cases are discussed.

When using DNA data from different people in order to investigate relationships between them, or more specifically to determine which *pedigree* connects them, certain types of locations in the DNA-strands, called *markers* or *loci* (singular: loci), are looked at. At these markers, each person has one out of several different variants, called *alleles*. We can draw conclusions about familial relationships from these markers since the alleles are transferred from parent to offspring. More specifically, we investigate *autosomal* markers, which have two alleles, one inherited from the mother and one inherited from the father. We will use the notation a/b to indicate that a certain individual has alleles a and b at a particular (autosomal) marker, sometimes we will say that a person has *genotype* a/b. The order of the letters will not be relevant, hence $b/a = a/b$. There are a few different types of markers one could look at. The most common type in forensic applications is called *short tandem repeat* (STR), see [10] or [2] for a detailed description. We will almost exclusively be concerned with STR markers.

In general, when making inference on familial relationships with DNA data, there are quite a few mechanisms that need to be modeled. We will not go through them all here but in [10], these are categorized into three different levels: *population level*, *pedigree level* and *observational level*. Population level models regard the treatment of founder alleles, i.e., how we should handle genotypes for persons in the pedigree with no parents. Observational level models try to account for complications that can occur in data collection, for example

measurement errors and mixtures (i.e., cases where measurements are performed on mixtures of DNA from several sources). Here, we will concentrate on pedigree level modeling, i.e., the transmissions of alleles between generations, and we will make simplifying assumptions regarding population and observational level modeling. For example, we will assume Hardy-Weinberg equilibrium, see [2], and flawless data collection.

There is a number of different softwares available to make inference on familial relationship cases from DNA data, for example the aforementioned Familias[1]. For a more thorough review, see [9].

4.1 Likelihood ratio computations from pedigrees

The hypotheses in familial relationship cases are statements that a person X is related to another person Y in a particular way. These statements are statements about the pedigree, hence we can formulate our hypotheses in terms of pedigrees and draw them in graphs. Let's consider an example.

4.1.1 A simple example

A man claims to be the father of a child and the mother is not available for genotyping. We are asked to assist in judging the credibility of the man's claim. We want to consider two hypotheses, (H_1) 'the man is the father of the child', and (H_2) 'the man is not the father of the child'. The pedigrees representing these hypotheses can be seen in Figure 4.1. Furthermore, we have data on one marker at which the putative father has alleles a/a and the child has alleles a/b.

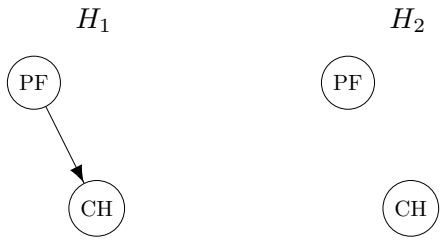

Figure 4.1: Two pedigrees representing two competing hypotheses. The putative father is denoted by PF and the child by CH.

The pedigrees in Figure 4.1 are of the simplest kind but there is no real limit of how complicated they can be in general. As in this example, it is not uncommon that one of the pedigrees is unconnected, representing a person (or

[1] http://familias.no/english/

a group of people) being unrelated to the rest of the members in the family tree. Our task now is to compute the likelihood ratio $\Pr(E|H_1)/\Pr(E|H_2)$, where the evidence, E, is the genotypes we have observed. The likelihoods themselves are probabilities that we observe these genotypes when choosing people at random from the population (given the corresponding pedigree). In order to compute such probabilities, we have to make comparisons towards data bases to see how common the observed alleles are among the population. More specifically, for each marker we assume knowledge of the *population frequency vector*, $\pi = (\pi_1, \ldots, \pi_n)$, which describes the frequencies of alleles among the relevant population, at that specific marker.

Now we return to the likelihood ratio computation. If we split up the evidence into E_{PF} and E_{CH}, which consists of the genotype of PF and CH respectively, we have that

$$\Pr(E|H_i) = \Pr(E_{PF}, E_{CH}|H_i) = \Pr(E_{CH}|E_{PF}, H_i)\Pr(E_{PF}|H_i)$$

for $i = 1, 2$. Since the genotype of PF has the same distribution under H_1 and H_2, we have that $\Pr(E_{PF}|H_1) = \Pr(E_{PF}|H_2)$. We can see that these factors will cancel each other out in the likelihood ratio. Under H_1, given that the father has genotype a/a, we know that he will transmit allele a to the child, hence the child's allele b must come from the mother (if we disregard the complications discussed in Section 4.2). The probability that an unknown mother will transmit allele b to CH is equal to π_b, which is then our value for $\Pr(E_{CH}|E_{PF}, H_1)$. On the other hand, under H_2 the genotype of CH is independent of PF, hence we draw them from the population frequency and obtain $\Pr(E_{CH}|E_{PF}, H_2) = 2\pi_a\pi_b$, where the factor 2 can be explained by the fact that the order of the alleles is irrelevant. We conclude that the likelihood ratio for H_1 against H_2 is $1/2\pi_a$.

4.1.2 Data on several markers

When using STR markers in real familial relationship cases, we usually have data on more than one marker, normally around 15. In general, the inheritance for these markers are not completely independent of each other, see Chapter 4 of [10]. However, assuming independence is somewhat standard and a rather good approximation for most marker sets, hence we will make this assumption from now on.

Suppose we are in the same case as in Section 4.1.1, with one putative father and one child, but that we have data on 15 markers instead of just one. We denote the evidence by $E = \{E_1, \ldots, E_{15}\}$ and, as usual, we want to compute the likelihood ratio $\Pr(E|H_1)/\Pr(E|H_2)$. Since the markers are assumed to be

independent, each of the likelihoods can be factorized as

$$\Pr(E|H_i) = \prod_{k=1}^{15} \Pr(E_k|H_i)$$

for $i = 1, 2$. This will make sure that we can make a similar factorization for the whole likelihood ratio, hence

$$LR = \frac{\Pr(E|H_1)}{\Pr(E|H_2)} = \prod_{k=1}^{15} LR_k = \prod_{k=1}^{15} \frac{\Pr(E_k|H_1)}{\Pr(E_k|H_2)}, \tag{4.1}$$

i.e., we simply have to repeat the calculations of Section 4.1.1 fifteen times.

4.1.3 Computations using Bayesian networks

It is not hard to see that pedigrees of Section 4.1.1 can be viewed as Bayesian networks. In spite of this, the computations performed in this section do not use the same method of analysis and the same terminology as is presented in Chapters 2 and 3. However, it is of course possible to use the theory presented in Chapter 2 to perform the computations in familial relationship cases as well, see for example [7], [8], [14] and [12].

Recall the pedigree representing H_1 in Figure 4.1. There is an edge pointing from PF to CH because, under H_1, PF is the father of CH. If we want to perform the analysis of Chapter 2 on this network, we need to define its different components in a more specific way. In particular, we need to specify the random variables that PF and CH should represent, i.e., we need to specify what values they can attain and we need to specify their CPDs. While this can be done directly on the graphs in Figure 4.1, each specification will be quite complex. If we have n possible alleles at the current marker, there are n^2 possible genotypes a person can have. To specify the distribution of the child's genotype given the genotype of the father, we need to specify $n^4 - n^2$ probabilities. Moreover, many of these probabilities are non-trivial to produce. Therefore, it is much more convenient to increase the number of nodes and let each genotype be represented by two nodes. For example, we could name the nodes PFMA, PFPA, CHMA and CHPA, for *putative father's maternal allele*, *putative father's paternal allele*, etc. The problem is now that we will never be able to have evidence on these nodes since we never know which of the alleles on an autosomal marker is paternal and which is maternal. Hence, we also want to add nodes for the genotypes of the people, we call them PFG and CHG, and these nodes will be determined by the paternal and maternal alleles in a deterministic way. In Figure 4.2 we present the final graph.

The CPDs needed to be specified in accordance with the graph in Figure 4.2 are straightforward to produce. The variables without parents are just drawn

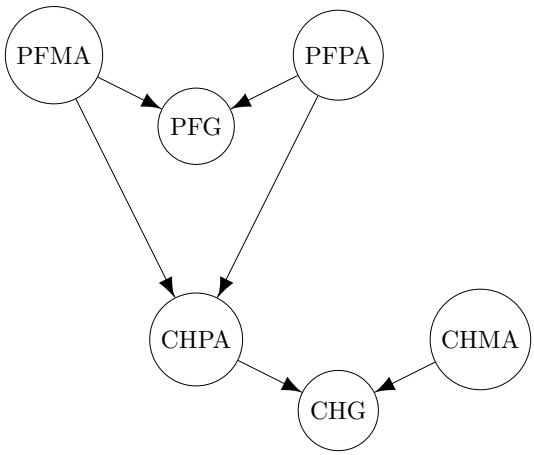

Figure 4.2: A Bayesian network graph that could be used to calculate the likelihood for H_1 in the example of Section 4.1.1.

from the population frequency π, and the genotype nodes are deterministically given by their parents. What remains is the node CHPA, which is equal to PFMA or PFPA with equal probability. Then we can apply the variable elimination algorithm in Chapter 2 to compute the desired likelihood. Naturally, we should produce a similar network for H_2 in order to finally be able to produce the likelihood ratio.

The general approach of using Bayesian network analysis to perform inference in DNA cases can handle quite a lot of complications and features. In Section 3.2, we already presented the idea of including a hypothesis node in a Bayesian network and this can of course be done here as well. In general, nodes can be added in many different ways to model difficulties on population, pedigree and observational levels. In [8], this is done neatly on a number of examples by locally isolating the mechanisms to be modeled. One of the most important such mechanisms is mutations.

4.2 Mutations

The calculations made to compute the likelihood ratio in Section 4.1.1 did not include many random elements. The only randomness in the transmission of alleles from one generation to the next was in the choice of whether the paternal or maternal allele is transferred to the child. This is an oversimplification of reality, partly because mutations might happen. An allele transmitted from a parent to its offspring might show up as another allele in the offspring. The following example shows that, although mutations are rare, we need to account for them in our calculations.

Example 3. Consider the same setup as in the example in Section 4.1.1 but instead of data on just one marker, we have data on 15 markers. The hypotheses are the same. On marker number 15, PF has alleles a/a and CH has alleles b/c, while on markers 1-14 the alleles of PF and CH agree in a similar way as in the example of Section 4.1.1, i.e., there is at least one match between the alleles of PF and CH. According to the discussion in Section 4.1.2, we want to compute each likelihood ratio LR_k separately. With the same way of reasoning as in Section 4.1.1, we have that

$$LR_k = \frac{\Pr(E_{CH,k}|E_{PF,k}, H_1)}{\Pr(E_{CH,k}|E_{PF,k}, H_2)}$$

where $E_{CH,k}$ and $E_{PF,k}$ denote the genotypes on marker k. Given that PF is the real father of CH and given that PF has genotype a/a, according to our reasoning so far, it is impossible for CH to have genotype b/c. Hence, we must have that $\Pr(E_{CH,15}|E_{PF,15}, H_1) = 0$, which implies that $LR_{15} = 0$ and $LR = 0$.

So can we make the indisputable conclusion that PF is not the real father of CH in this example? Even though our calculations suggest it, this is not what should be done since mutations might have occurred. In fact, agreeing alleles on 14 out of 15 markers is usually quite strong evidence that PF is the real father of CH. So in order for LR computations to be useful, they need to take the possibility of mutations into account.

A mathematical model for the mutation process should for each allele i specify the probability of mutation to each other allele j. It is not hard to make the conclusion that we can view this as a time homogeneous finite state Markov chain. If we denote the mutation probabilities by m_{ij} and the transition matrix for the corresponding Markov chain by M, we have

$$M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{pmatrix}.$$

In this context, we will call such a matrix a *mutation matrix* and it completely specifies the mutation process. The diagonal elements m_{ii} are the probabilities of no mutation from the corresponding allele, hence $1 - m_{ii}$ is the probability of mutation from allele i.

The elements of mutation matrices are not directly estimated from data and there are a few reasons for this. Firstly, for some STR markers there are more than 70 possible alleles, hence there are more than $70^2 \approx 5\ 000$ parameters to estimate. To make things worse, the probability of mutation is very small,

usually smaller than 0.005, and the probability of seeing a specific mutation, from an allele a to another specified allele b, is even smaller. In many cases there are no observations of such mutations, hence the frequency estimate would be zero. Moreover, in many cases there is no way of knowing for sure that a mutation has (not) occurred. For example if a father with a/b and a mother with a/b have a child with a/a, even though unlikely, one of the parents could have transmitted a mutated b allele to the child. However, we would 'observe' this particular occurrence as a non-mutation case.

So instead of estimating mutation matrices using frequencies, we settle for parametric models motivated by a mixture of biological knowledge, simplicity and mathematical tractability. In Paper I, we present the most common mutation models.

It turns out that even when we are able to construct a biologically reasonable mutation model, we can still run into problems. To see an example of this, we redo the computations in Example 3 by using a mutation matrix M. As one would guess, we will not get $LR_{15} = 0$ now. We have that

$$\begin{aligned}
&\Pr(E_{CH,15}|E_{PF,15}, H_1) = \\
&= \Pr(\text{Mutation from } a \text{ to } b) \Pr(\text{CH maternal allele is } c) + \\
&+ \Pr(\text{Mutation from } a \text{ to } c) \Pr(\text{CH maternal allele is } b) = \\
&= m_{ab}\pi_c + m_{ac}\pi_b.
\end{aligned}$$

The other likelihood is not affected by mutations at all since H_2 contains no inheritances. We get that $\Pr(E_{CH,15}|E_{PF,15}, H_2) = 2\pi_b\pi_c$, hence

$$LR_{15} = \frac{m_{ab}\pi_c + m_{ac}\pi_b}{2\pi_b\pi_c}. \tag{4.2}$$

Recall the pedigrees in Figure 4.1 which we used to represent the hypotheses H_1 and H_2. Now we will consider an alternative way of formulating these pedigrees.

Example 4. Again, we have the same setup as in the example in Section 4.1.1. However, here we will use the pedigrees in Figure 4.3 to represent H_1 and H_2. The idea is that, instead of drawing the child's maternal allele from the population frequency π, we add an unknown mother in the pedigree whose alleles we draw from π, and then use the mutation model to obtain the child's maternal allele. The derivations made to produce (4.2) are the same in this case, however, $\Pr(\text{CH maternal allele is } i)$ has changed. Let λ be the distribution of the child's maternal allele, i.e., $\lambda = \pi M$. Then we will end up with

$$LR_{15} = \frac{m_{ab}\lambda_c + m_{ac}\lambda_b}{\pi_b\lambda_c + \pi_c\lambda_b} \tag{4.3}$$

which will in general not be equal to (4.2).

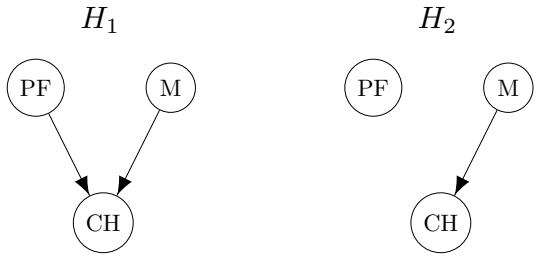

Figure 4.3: An alternative way to represent the hypotheses in the example in Section 4.1.1 using pedigrees.

Now we should be curious and ask if it is a problem that the two approaches to the same problem yield different results, and if so, why is it a problem? Well, there is no convincing argument about which of the two sets of pedigrees, the ones in Figure 4.1 or the ones in Figure 4.3, should be preferred. Both of them perfectly explain the hypotheses we are interested in and there are even different practices among the established softwares regarding the inclusion of untested parents. Even though the difference between (4.2) and (4.3) might be small in most cases, the final LR is a product of these individual LRs, hence the difference could easily be magnified. Moreover, the people that would interpret the results, for example the parties in a paternity dispute, usually have no insight in the computation of the likelihood ratio. Therefore, the discrepancy in the answers, how small it may be, could undermine the authority of the used software. If two seemingly equivalent formulations give rise to different answers, which of them, if any, should be trusted?

The discussion regarding Example 4, and in particular equations (4.2) and (4.3), suggests that the problem is avoided if $\lambda = \pi$, i.e., if $\pi M = \pi$. Hence, we are looking for mutation matrices whose stationary distribution is equal to the allele population frequency. In Paper I, we call such mutation matrices *stationary* and we discuss a number of different approaches to produce stationary mutation matrices.

Chapter 5

Summary of papers

Paper I

In this paper, we discuss the necessity for stationary mutation matrices and present a few popular ways for modeling the mutation process, some of which give rise to stationary mutation matrices. However, we believe that it is important to keep focus on the biological soundness of the mutation model. Stationarity should not by default supercede the aim to model the real mutation process in a reasonable way. Therefore, we propose a method of *stabilizing* mutation matrices. The idea is to start with an existing and biologically reasonable mutation matrix, M, and change it as little as possible to obtain a stationary mutation matrix S, i.e., we want $\pi S = \pi$, where π is the population frequency vector. The new mutation matrix S is then called a *stabilization* of M and the goal is that S inherits the properties from M that makes it biologically reasonable.

Now we have to decide on what we mean by changing a matrix "as little as possible". In this paper we do this by comparing matrices element-wise and considering how much exchanging two matrices would alter likelihood ratio calculations for some common pedigrees. We conclude that a reasonable measure of closeness between mutation matrices is

$$f_{ratio}(M,S) = \max\left\{\max_{i,j}\left\{\frac{m_{ij}}{s_{ij}}\right\}, \max_{i,j}\left\{\frac{s_{ij}}{m_{ij}}\right\}\right\}. \tag{5.1}$$

So given a matrix M, we are looking for a matrix S fulfilling $\pi S = \pi$ and minimizing f_{ratio}.

It turns out that it is necessary to impose more restrictions on a stabilization. In particular, it is often necessary to have better control of the diagonal elements of stabilizations. Recall that the diagonal elements in a mutation matrix specify the probability of non-mutation and we want them to be quite a lot larger than

the off-diagonal elements. We suggest a few different mechanisms for controlling the size of the diagonal elements. These mechanisms give rise to three different stabilization methods which we define in this paper. We also provide theoretical results for their existence and we use them on data. The results are mixed and for some markers we have to make the conclusion that stationary mutation models are not recommendable.

Paper II

In this paper, we focus on the variable elimination algorithm and which classes of BNs it can be used for. We describe how it can be implemented on Gaussian BNs using the canonical forms of Definition 6. Inspired by this, we propose a new class of networks constructed by adding a gamma distributed variable to model the precision of the Gaussian variables in a Gaussian BN. Here, we make the restriction that the precision of *all* Gaussian variables in the network needs to be modeled by the gamma variable.

As described in Chapter 2, when implementing the VE algorithm for a specific class of BNs, it is important to identify the appropriate factor set. The BNs we consider in this paper give rise to factors in the form (2.6) which we call *Γ-canonical forms*. We also present how the local operations work on these factors but it turns out that the Γ-canonical forms are not quite closed under the local operations. One important result in this paper is that marginalizing the gamma variable τ out of a Γ-canonical form will result in a factor that is proportional to the (multivariate) Student-t distribution. In fact, this marginalization is the only local operation that will result in something else than a Γ-canonical form, i.e., using the language of Paper III, it is the only operation that will take us out of the prefamily. We also apply the theory presented in this paper to the glass data described in Section 3.2.

Paper III

In this paper, we further extend the classes of BNs for which we can perform exact inference. The usual problem that occurs when one tries to implement the variable elimination algorithm for a specific class of BNs is the difficulty of defining a factor set that is closed under factor marginalization. This insight led us to define *prefamilies*, i.e., factor sets that are not necessarily closed under marginalization, and to construct an algorithm that performs variable elimination on prefamilies. We present the *prefamily variable elimination* algorithm, which is a recursive version of the VE algorithm that can be applied to prefamilies. The recursive nature of this algorithm allows us to use numerical integration whenever marginalization results in a factor outside the prefamily.

The idea of the prefamily variable elimination algorithm came up when working with Gaussian BNs and trying to model the precision of Gaussian variables in a more flexible way than in the rather restrictive networks of Paper II. We demonstrate the prefamily variable elimination by implementing it on this class of networks, which we in this paper call Γ-Gaussian BNs. Algorithm 4 of this paper describes our implementation of the algorithm that is meant to be applied to Γ-Gaussian BNs. This implementation contains a few tricks that are helpful in specific situations.

Another important contribution is the handling of finite variables. Including finite variables in an otherwise continuous network will not alter the possibility for performing exact inference, given that no finite variables have continuous parents. This is well known and is usually presented together with variable elimination on Gaussian BNs. In this paper we present this extension in a general way. We keep track of what happens with the appearance of the corresponding factor set and we prove that the (pre)family property is preserved under this extension.

Bibliography

[1] Colin Aitken and Franco Taroni. *Statistics and the evaluation of evidence for forensic scientists*, volume 16. Wiley Online Library, 2004.

[2] John Buckleton, Christopher M Triggs, and Simon J Walsh. *Forensic DNA evidence interpretation*. CRC press, 2005.

[3] John M Butler. *Advanced topics in forensic DNA typing: interpretation*. Academic Press, 2014.

[4] Roger Cook, Ian W Evett, Graham Jackson, PJ Jones, and JA Lambert. A hierarchy of propositions: deciding which level to address in casework. *Science & Justice*, 38(4):231–239, 1998.

[5] Robert G Cowell, A Philip Dawid, Steffen L Lauritzen, and David J Spiegelhalter. *Probabilistic networks and expert systems*. Springer-Verlag, 1999.

[6] JM Curran, CM Triggs, JR Almirall, JS Buckleton, and KAJ Walsh. The interpretation of elemental composition measurements from forensic glass evidence: Ii. *Science & Justice*, 37(4):245–249, 1997.

[7] A Philip Dawid, Julia Mortera, Vincenzo L Pascali, and D Van Boxel. Probabilistic expert systems for forensic inference from genetic markers. *Scandinavian Journal of Statistics*, 29(4):577–595, 2002.

[8] A Philip Dawid, Julia Mortera, and Paola Vicard. Object-oriented bayesian networks for complex forensic dna profiling problems. *Forensic Science International*, 169(2):195–205, 2007.

[9] Jiří Drábek. Validation of software for calculating the likelihood ratio for parentage and kinship. *Forensic Science International: Genetics*, 3(2):112–118, 2009.

[10] Thore Egeland, Daniel Kling, and Petter Mostad. *Relationship Inference with Familias and R: Statistical Methods in Forensic Genetics*. Academic Press, 2016.

[11] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[12] Julia Mortera, A Philip Dawid, and Steffen L Lauritzen. Probabilistic expert systems for dna mixture profiling. *Theoretical population biology*, 63(3):191–205, 2003.

[13] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[14] Franco Taroni, Colin Aitken, Paolo Garbolino, and Alex Biedermann. *Bayesian Networks and Probabilistic Inference in Forensic Science*. John Wiley & Sons, Ltd, 2006.

[15] Grzegorz Zadora, Agnieszka Martyna, Daniel Ramos, and Colin Aitken. *Statistical Analysis in Forensic Science: Evidential Values of Multivariate Physicochemical Data*. John Wiley & Sons, 2013.